What is Claimed:

1. A method of marking for authentication a computer program with a bit-string pattern, the method comprising the steps of:

a) generating a bit-string pattern, each bit in the bit-string pattern having a binary value and at least one bit having a first value;

b) searching for at least one polymorphic statement in the computer program;

c) associating the one bit having the first value with the polymorphic statement found in step (b); and

d) altering the polymorphic statement;

wherein altering the polymorphic statement marks the computer program.

2. The method of claim 1 wherein generating the bit-string pattern includes generating multiple bits having first and second values;

associating each of the multiple bits with a polymorphic statement;

modifying a polymorphic statement corresponding to a bit having a first value; and

leaving unmodified a polymorphic statement corresponding to a bit having a second value.

3. The method of claim 1 including the step of:

e) providing a pointer for locating a statement in the computer program; and

searching for the one polymorphic statement in step (b) includes searching for the one polymorphic statement based on the statement located by the pointer.

4. A method of marking, for authentication, source code of a computer program, designated as P, and having a complied version of the computer program, designated as E, the method comprising the steps of:

a) generating a binary bit-string pattern, designated as B, having a predetermined value;

b) modifying P to produce a separate program P1, such that the separate program P1, when compiled, functions identically to P;

wherein modifying P includes one of the following steps:

i) modifying inline assembly code of P based on B; and

ii) manipulating binary executable code of E based on B.

5. The method of claim 4 wherein step (a) generates a binary bit-string pattern having a value not equal to zero.

6. The method of claim 4 wherein step (i) includes correlating a binary bit in B to at least one statement of inline assembly code of P, and

modifying the one statement when the binary bit has a first value.

7. The method of claim 6 wherein the first value of the binary bit is 1.

8. A method of marking for authentication a computer program with a bit-string pattern, the bit-string pattern including a plurality of values, the method comprising the steps of:

a) associating the plurality of values with a respective plurality of predetermined computer statements, in which each predetermined computer statement is expressible as first and second equivalent operations;

b) selecting a first non-processed value of the bit-string pattern;

c) searching the computer program for at least one predetermined computer statement corresponding to the first non-processed value of the bit-string pattern;

d) expressing the predetermined computer statement found in step (c) as one of the first and second equivalent operations;

e) marking the first non-processed value of the bit-string pattern as processed; and

f) repeating steps (b) through (e) for each non-processed value of the bit-string pattern.

9. The method of claim 8 in which step (d) includes altering the predetermined computer statement from the first operation to the second operation, if the first non-processed value of the bit-string pattern is a value of 1.

10. The method of claim 8 including the step of:

generating the bit-string pattern having multiple bits of first and second values and a bit length smaller than or equal to a number of predetermined statements in the computer program.

11. The method of claim 8 including the step of:

g) providing a pointer for locating a predetermined statement in the computer program; and

searching the computer program of step (c) includes searching for the predetermined statement located by the pointer in step (g).

12. A method of authenticating a second computer program against a first computer program, the method comprising the steps of:

a) accessing a bit-string pattern, each bit in the bit-string pattern having a binary value and at least one bit having a first value;

b) searching for at least one polymorphic statement in the second computer program;

c) associating the one bit having the first value with the polymorphic statement found in step (b);

d) altering the polymorphic statement in the second computer program;

e) comparing the polymorphic statement in the second computer program, after altering the polymorphic statement in step (d), against a corresponding polymorphic statement in the first computer program; and

f) determining that the second computer program is a modified version of the first computer program, if the polymorphic statements compared in step (e) are not similar.

13. The method of claim 12 including the steps of:

(g) associating another bit having a second value with another polymorphic statement found in step (b);

(h) comparing the other polymorphic statement of step (g) against a corresponding polymorphic statement in the first computer program; and

(i) determining that the second computer program is a modified version of the first computer program, if the polymorphic statements compared in step (h) are not similar.

14. The method of claim 13 including the step of:

repeating steps (b) through (f) for another bit in the bit-string pattern having a first value.

15. The method of claim 13 including the step of:

repeating steps (g) through (i) for another bit in the bit-string pattern having a second value.

16. A method of authenticating a second computer program against a first computer program, the method comprising the steps of:

a) accessing a first bit-string pattern, each bit in the first bit-string pattern having a binary value associated with a corresponding polymorphic statement in the first computer program;

b) searching for at least one polymorphic statement in the second computer program;

c) assigning a bit having a binary value to the polymorphic statement found in step (b);

d) assigning another bit having a binary value to another polymorphic statement found in step (b);

e) generating a second bit-string pattern including the bits assigned in steps (c) and (d); and

f) determining that the second computer program is a modified version of the first computer program, if the second bit-string pattern does not match the first bit-string pattern.